

Random walks on graphs with interval weights and precise marginals

Damjan Škulj

University of Ljubljana, Slovenia
damjan.skulj@fdv.uni-lj.si

September 22, 2015

Abstract

We propose a model of random walks on weighted graphs where the weights are interval valued, and connect it to reversible imprecise Markov chains. While the theory of imprecise Markov chains is now well established, this is a first attempt to model reversible chains. In contrast with the existing theory, the probability models that have to be considered are now non-convex. This presents a difficulty in computational sense, since convexity is critical for the existence of efficient optimization algorithms used in the existing models. The second part of the paper therefore addresses the computational issues of the model. The goal is finding sets of weights which maximize or minimize expectations corresponding to multiple steps transition probabilities. In particular, we present a local optimization algorithm and numerically test its efficiency. We show that its application allows finding close approximations of the globally best solutions in reasonable time.

Keywords

weighted graph, random walk, Markov chain, imprecise Markov chain, reversible Markov chains, local optimization, global optimization

1 Introduction

1.1 Modelling uncertainty in Markov chains and weighted graphs

Markov chains with the property that every sequence of states is equally likely no matter whether the process runs forwards or backwards are said to be *reversible*. Reversible Markov chains are often interpreted and modelled with *random walks* on weighted graphs ([1, 3, 4, 10, 11, 13, 21]) where the states of the chain are the vertices of the graph and transition probabilities are proportional to the weights of the edges incident to the initial vertex. Reversible Markov chains are often used in Monte Carlo methods ([14, 16, 17]). Random walks on graphs have become very popular in network analysis ([8, 19, 20, 29]), social networks ([7, 18, 23, 28]) and web recommender systems ([12]).

Modelling real world phenomena with Markov chains requires estimating a large number of parameters. Even with ever growing amounts of data at disposal this task is often impossible to achieve without serious uncertainty in the estimates. Ignoring this fact and regarding the parameters as precise leads to overprecise unreliable results. The need for more robust models for probability has led to various models known under the common name as *theory of imprecise probabilities* ([5]). In particular, for Markov chains the theory of *imprecise Markov chains* has been developed for discrete ([9, 15, 25]) as well as continuous case ([26]). Most of the existing models are based on the theory of lower previsions ([22]).

In the core of the theory of imprecise Markov chains is the idea that transition probabilities at each step are modelled with convex sets of probability distributions rather than single transition probabilities. Equivalently, all relevant probability distributions can then be modelled by non-additive functionals called *coherent lower previsions*, which are defined as lower envelopes of sets of additive functionals.

Weights in graphs often also reflect some relation between vertices obtained on the basis of imperfect data. One way of expressing the resulting uncertainty is to use intervals instead of precise weights. While being a compelling generalisation, the related optimisation problems seem to be generally hard ([2, 6]). Up until now finding minimum spanning tree and shortest paths in graphs with weighted intervals have received a lot attentions, while random walks seem to have not yet been explored. The lack of appropriate models of imprecise Markov chains and apparent high complexity of the general model might be among the reasons for this. The high complexity is also the main reason for our decision to keep our model simple by not allowing weights to vary completely freely within interval bounds, but instead assuming the sum of weights of edges incident to a given vertex to be constant. This could only be efficiently achieved by allowing loops, which then contain the non-allocated weight mass.

1.2 Model

The aim of the present article is to extend the theory of imprecise Markov chains for the case of reversible chains; more specifically, random walks on weighted graphs with interval weights. Interval weights are interpreted as sets containing the precise weights that will actually set the probabilities of transitions. We also assume that weights are not constant in time but rather at every time step an unknown mechanism selects new set of weights, for which, except that they belong to the given intervals, we have no information available. Once the weights at certain time step are selected, transitions are calculated in the usual way. In our model we restrict the set of weights by requiring that total sum of weights of edges incident to a vertex is constant and precisely known. This is achieved by assigning the remaining mass to the loops (i.e. edges connecting the same vertices). This restriction will allow an efficient local optimisation for calculation of multiple steps probability bounds. Actually a similar effect is the result of the rate of leaving a state when modelling continuous time Markov chains. Having precisely given marginals while dependencies are imprecise is not that uncommon since usually there is a lot more data available about marginal values than about dependencies.

In comparison with the existing models of imprecise Markov chains the most

important differences are that probability models behind our model are not necessarily convex and that in general they do not satisfy Bellman’s principle of optimality (see e. g. [24]). Consequently calculating bounds for multiple steps transition probabilities is a much more computationally intensive task.

We give the detailed description of the model in Sections 2 and 3.

1.3 Results

While our theoretical model is not very different from other models of imprecise Markov chains, there are substantial differences when it comes to computations. We will investigate computing n -steps transition probabilities, which are the basis for any analysis with Markov chains. As imprecision is involved, we cannot speak about single precisely given transition probabilities, but rather their lower and upper bounds. Moreover, in the case of imprecise probabilities, bounds for elementary events are not sufficient to specify the corresponding probability models. Therefore we have to consider computing bounds for more general expectations.

The existing models of imprecise Markov chains allow setting transitions from one state to others independently from one another. This ensures convexity of the underlying probability models and possibility to apply Bellman’s principle of optimality. These properties then imply existence of a single local and therefore also global optimum, which is found by sequentially maximizing expectations via linear programming. Complexity of the problem thus remains linear in the number of time steps. The problem of finding extremal expectation in our settings becomes considerably more complicated. In general the problem is not convex and neither it satisfies Bellman’s principle. Consequently, in general multiple local optima exist, and that backwards induction is not applicable. This means that irreducible dimensionality of the problem grows exponentially with the number of time steps.

Our main numerical result is a local optimisation algorithm which we propose in Section 4. Given an initial weight function it returns a local optimal solution. Global extrema, though, are still sought by taking various starting points and do local optimisation. As the size of the space of all feasible points is far to big to be tractable by any reasonable computer, we cannot provide a criterion that would definitely ensure that obtained solution is global maximum. But numerical testing shows that in most cases a reasonable approximation of global solution can be obtained by taking a moderate number of starting points. Even more convincingly it shows that if weight functions were chosen by random, without applying the local optimization, then it would almost certainly take incomparably larger samples to get results comparably close to the optimal solution. While, as far as we are aware, no other algorithms exist for optimization of random walks on graphs with interval weights, we can only compare our method to random choice, which is therefore by far outperformed.

2 Model settings

Let \mathcal{X} be a non empty set of *states*. We will usually denote the number of states by s . We consider random walks on the graph with vertices \mathcal{X} and weighted edges that are given in the form of an interval *weight function*. The probabilities

of transitions between states are assumed to be proportional with the weights. More precisely, if $w: \mathcal{X}^2 \rightarrow \mathbb{R}_{\geq 0}$ is a weight function then

$$P_w(x, y) = P_w(x_{n+1} = y | X_n = x) = \frac{w(x, y)}{\sum_{y \in \mathcal{X}} w(x, y)} = \frac{w(x, y)}{W(x)}. \quad (1)$$

In this paper we assume that the denominator in the above fraction is a fixed function of the state. That is, we assume a precise function $W: \mathcal{X} \rightarrow \mathbb{R}_+$ as a sum of weights edges incident to a vertex. This restriction will enable us to obtain an efficient optimization algorithm. When modeling uncertainty we often have fairly good information on the long term distributions over the set of states (which are closely related to the corresponding weights) but much less certainty regarding the transition probabilities. We will thus allow weights $w(x, y)$ where $x \neq y$ to vary freely within given intervals and the remaining weight mass will be used to model the loop weight $w(x, x)$.

Formally, we define a set

$$\mathcal{W} = \left\{ w: \mathcal{X}^2 \rightarrow \mathbb{R}_{\geq 0}: \underline{w}(x, y) \leq w(x, y) \leq \bar{w}(x, y), \sum_{y \in \mathcal{X}} w(x, y) = W(x) \right\}.$$

where \underline{w} and \bar{w} are arbitrary such that

$$\underline{w} \leq \bar{w} \quad \text{and} \quad \sum_{\substack{y \in \mathcal{X} \\ y \neq x}} \bar{w}(x, y) \leq W(x).$$

Every weight function in \mathcal{W} defines transition probabilities via equation (1). Our aim is to provide some basic properties of the corresponding Markov chains.

Additionally, to avoid problems with uniqueness of the invariant distributions, we will assume that for all pairs of states

$$\text{either } \bar{w}(x, y) = 0 \text{ or } \underline{w}(x, y) > 0 \quad (2)$$

and that there is a path between every pair of states consisting of edges with strictly positive weights.

3 Imprecise Markov chains

3.1 Transition operators with separately specified rows

Markov chains whose parameters are only partially known have been studied recently into some details under the name *imprecise Markov chains* ([9, 25]) or *Markov set chains* ([15]). Here we give basic ideas and notations related to the theory described in [9].

An *imprecise Markov chain* is a sequence $(X_n)_{n \in \mathbb{N} \cup \{0\}}$ of random variables taking values in a finite state space \mathcal{X} . The imprecise distribution corresponding to some X_n is given in the form of a set of probability distributions \mathcal{M}_n consisting of distributions compatible with the given partial knowledge of the process. In the case where \mathcal{M}_n are convex, they can be equivalently described in terms of lower expectation functionals

$$\underline{E}_n(f) = \min_{q \in \mathcal{M}_n} \sum_{x \in \mathcal{X}} q(x) f(x), \quad (3)$$

where $q: \mathcal{X} \rightarrow \mathbb{R}$ are probability mass functions corresponding to distributions in \mathcal{M} and f an arbitrary real valued map on \mathcal{X} .

The transition law between states is also given in imprecise way. That is by assuming a set of transition operators \mathcal{T} , which is called an *imprecise transition operator*. The following relation then holds

$$\mathcal{M}_{n+1} = \mathcal{M}_n \mathcal{T} = \{qT: q \in \mathcal{M}_n, T \in \mathcal{T}\}. \quad (4)$$

We do not assume transition probabilities being constant in time but only that they belong to the specified set of transition operators. Thus, an imprecise Markov chain is in principle time inhomogeneous with fixed constraints on transition probabilities.

Moreover, an imprecise transition operator \mathcal{T} maps some $f \in \mathbb{R}^{\mathcal{X}}$ to a set $\mathcal{T}f = \{Tf: T \in \mathcal{T}\}$. An imprecise transition operator \mathcal{T} is said to have *separately specified rows* if for every $T, T' \in \mathcal{T}$ there exists \tilde{T} so that $\tilde{T}_{ij} = T_{ij}$ for every j and $i \neq k$ and $\tilde{T}_{kj} = T'_{kj}$ for every j . That is the k th row can be chosen independently from the choice of other rows.

Now we have the following important property. If an imprecise transition operator \mathcal{T} has separately specified rows then there exist the minimal and maximal elements in the set $\mathcal{T}f$ for every $f \in \mathbb{R}^{\mathcal{X}}$, denoted by $\underline{T}f$ and $\bar{T}f$. The mappings $\underline{T}: f \mapsto \underline{T}f$ and $\bar{T}: f \mapsto \bar{T}f$ are called the lower and the upper transition operators respectively, and their values can be calculated via linear programming. Let \underline{E}_0 be an initial lower expectation operator and \underline{T} a lower transition operator. The expectation of $f(X_n)$, where $f \in \mathbb{R}^{\mathcal{X}}$ is calculated by repeatedly applying \underline{T} using linear programming and finally apply \underline{E}_0 , again via linear programming:

$$\underline{E}_n(f) = \underline{E}_0(\underline{T}^n f). \quad (5)$$

The above equation generalizes calculation of n -step transition probabilities, since, for instance,

$$P(X_n = y | X_0 = x) = E_x(\underline{T}^n 1_{\{y\}}) = \underline{T}^n 1_{\{y\}}(x),$$

where 1_A denotes the indicator function of the set $A \subseteq \mathcal{X}$. While in the case of precise Markov chains n -step transition probabilities between single states completely determine the distribution of X_n , in the case of imprecise transitions expectations (5) must be used instead (see [9] or [25] for more details).

3.2 Transition operators on weighted interval graphs

The transition operator with respect to a weight function w is a map $T_w: \mathbb{R}^{\mathcal{X}} \rightarrow \mathbb{R}^{\mathcal{X}}$ that is defined with

$$T_w f(x) = \sum_{y \in \mathcal{X}} P_w(x, y) f(y) = \sum_{y \in \mathcal{X}} \frac{w(x, y)}{W(x)} f(y). \quad (6)$$

Similarly we can define the action of T_w from the right by

$$qT_w(y) = \sum_{x \in \mathcal{X}} q(x) P_w(x, y) = \sum_{x \in \mathcal{X}} q(x) \frac{w(x, y)}{W(x)}. \quad (7)$$

In particular, if q is a probability mass function corresponding to X_n on the set of states then qT_w is the probability mass function corresponding to X_{n+1} .

We will stick to our general assumption that transition operators are (non-specified) function of time, rather than being constant in time. We now extend naturally both operators to vectors of weight functions $\mathbf{w} = (w_1, \dots, w_n)$ with

$$T_{\mathbf{w}}f = T_{w_1} \dots T_{w_n}f \quad (8)$$

and

$$qT_{\mathbf{w}} = qT_{w_1} \dots T_{w_n}. \quad (9)$$

Given a set of weights \mathcal{W} , we define an imprecise transition operator $\mathcal{T} = \{T_w : w \in \mathcal{W}\}$. It is clear that so defined imprecise transition operator does not possess the separately specified rows property. In fact there is no way to involve such a property, because of the symmetry of weights, that is the entry $w(x, y)$ which determines transition probability from x to y also determines the reverse transition probability. Yet those two belong to different rows of the corresponding transition matrix, namely, the first one in the row corresponding to x and the second one to the row corresponding to y . As a consequence, the notion of an upper or lower transition operators does not make sense here, as there is no unique maximal or minimal elements in the set $\mathcal{T}f := \{T_w f : T_w \in \mathcal{T}\}$. Moreover, while sets of the form \mathcal{T} or $\mathcal{T}f$ are convex, this is not any more the case with more general sets, such as $\mathcal{T}^2 := \{T_{w_1}T_{w_2} : w_1, w_2 \in \mathcal{T}\}$ or $\mathcal{T}^2 f$. This also means that optimization methods based on linear programming that are successfully applied in the theory of imprecise Markov chains cannot be applied on our case.

An imprecise Markov chain is said to be *regular* if there exists some positive integer r such that all transition operators in \mathcal{T}^n , where $n \geq r$, have all elements positive. According to our convention (2) every state is reachable from any other state, and since loops are also possible with strictly positive probability the chain is acyclic and therefore regular. It follows then (see [27]) that there exists the unique invariant set \mathcal{M} of probability distributions. Assuming fixed marginals, it follows that in our case the marginal distribution π , where $\pi(x) = \frac{W(x)}{W}$ is the common unique distribution corresponding to all operators in \mathcal{T} . Thus, $\{\pi\}\mathcal{T} = \{\pi\}$, which implies that $\{\pi\}$ is the unique invariant set of distributions.

3.3 Reversibility

One of the most important properties of Markov chains that can be represented as random walks on graphs is that they are reversible processes. That means that we have equal probability to observe a sequence of states if their order is reversed:

$$P(X_1 = x_1, \dots, X_n = x_n) = P(X_1 = x_n, \dots, X_n = x_1), \quad (10)$$

assuming that $(X_1 = x_1) = \pi(x_1)$, where π is the unique invariant distribution. Applying the above property to the case of $n = 2$ we obtain the *detailed balance condition*:

$$\pi(x)P(x, y) = \pi(y)P(y, x), \quad (11)$$

where $P(x, y)$ is the transition probability between x and y .

Clearly, a precise random walk on a graph with weight function w satisfies the detailed balance condition, due to the fact that

$$P(X_1 = x, X_2 = y) = \frac{w(x, y)}{W} = P(X_1 = y, X_2 = x),$$

where W denotes the sum of weights for all edges. The symmetry of interval weights also clearly implies that the lower probabilities $\underline{P}(X_1 = x, X_2 = y)$ and $\underline{P}(X_1 = y, X_2 = x)$ are the same and equal to $\frac{w(x, y)}{W}$.

Similarly we can calculate probabilities of the elementary events for more consecutive steps. Thus, for instance

$$\underline{P}(X_1 = x_1, \dots, X_n = x_n) = \frac{\prod_{i=1}^{n-1} w(x_i, x_{i+1})}{W \prod_{j=2}^{n-1} W(x_j)}, \quad (12)$$

which again is equal to the lower probability of the reversed sequence of states.

4 Numerical calculations

4.1 Calculating expectations bounds

Although we have a very simple expression (12) that allows calculating the lower probabilities of sequences of states it cannot be used directly to find lower (or upper) probabilities of more general events. An example would be calculating the 2-step lower probability of transition from x to y . In the precise case knowing the probabilities of all chains of states of length 3 would allow calculating such probability:

$$P(X_2 = y | X_0 = x) = \sum_{z \in \mathcal{X}} P(X_2 = y | X_1 = z) P(X_1 = z | X_0 = x).$$

However, the above formula is incorrect if lower probability \underline{P} is taken instead of P . The reason is that lower probabilities are in general *non-additive*. In our case, for instance, the lower probability $\underline{P}(x, y) = P_w(x, y)$ for some particular weight function w , but $\underline{P}(x, y') = P_{w'}(x, y')$ for another weight function w' , and there is usually no weight function that would induce the lower probabilities simultaneously.

What we need to find in the case of minimizing the 2-step transition probabilities is

$$\underline{P}(X_2 = y | X_1 = x) = \min_{w_1, w_2 \in \mathcal{W}} \sum_{z \in \mathcal{X}} P_{w_1}(x, z) P_{w_2}(z, y). \quad (13)$$

A more general problem is to find the bounds for the expectation of some function $f(X_n)$ given the information that $X_0 = x$, or some probability distribution of X_0 over the set of states. But even if this seems like an unnecessarily more complex problem, we would in fact not gain much in terms of simplicity by restricting to n -step transition probabilities alone.

A (precise) probability distribution over \mathcal{X} can be described via probability mass function (pmf) $q: \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ where $q(x) = P(X_0 = x)$. The expectation of some $f \in \mathbb{R}^{\mathcal{X}}$ with respect to q is the scalar product

$$\langle q, f \rangle := E_q(f) = \sum_{x \in \mathcal{X}} q(x) f(x). \quad (14)$$

To extend the above formulation for more general case of the expectation after n steps, we consider the vector of weight functions $\mathbf{w} = (w_1, \dots, w_n)$. Thus at

k -th step the weight function w_k is assumed to induce the transitions. We will then denote

$$\langle q, f \rangle_{\mathbf{w}}^n := \langle q, T_{\mathbf{w}} f \rangle. \quad (15)$$

Our goal is to find

$$\underline{\langle q, f \rangle}^n = \min_{\mathbf{w} \in \mathcal{W}^n} \langle q, f \rangle_{\mathbf{w}}^n \quad \text{and} \quad \overline{\langle q, f \rangle}^n = \max_{\mathbf{w} \in \mathcal{W}^n} \langle q, f \rangle_{\mathbf{w}}^n. \quad (16)$$

Since, clearly, $\overline{\langle q, f \rangle}^n = -\langle q, -f \rangle^n$, we only need to consider the minimization version. Allowing arbitrary real valued functions q instead of restricting to probability mass functions does not change anything in the sense of problem complexity. Therefore we will from now on assume q and f to be arbitrary real valued functions.

Proposition 1. *Let f and $q \in \mathbb{R}^{\mathcal{X}}$ and $\mathbf{w} \in \mathcal{W}^n$. Then $\langle q T_{\mathbf{w}}, f \rangle = \langle q, T_{\mathbf{w}} f \rangle$.*

Proof. We proceed by induction on the length n of vector \mathbf{w} , where the case $n = 0$ is trivial. Take now

$$\langle q T_{\mathbf{w}}, f \rangle = \langle q T_{w_1}, T_{(w_2, \dots, w_n)} f \rangle$$

denote $\tilde{f} = T_{(w_2, \dots, w_n)} f$ and continue with

$$\begin{aligned} &= \sum_{y \in \mathcal{X}} q T_{w_1}(y) \tilde{f}(y) \\ &= \sum_{y \in \mathcal{X}} \sum_{x \in \mathcal{X}} q(x) \frac{w_1(x, y)}{W(x)} \tilde{f}(y) \\ &= \sum_{x \in \mathcal{X}} q(x) \sum_{y \in \mathcal{X}} \frac{w_1(x, y)}{W(x)} \tilde{f}(y) \\ &= \langle q, T_{w_1} \tilde{f} \rangle = \langle q, T_{\mathbf{w}} f \rangle. \end{aligned}$$

□

Clearly the following holds.

Proposition 2. *The mapping $(q, w, f) \mapsto \langle q, f \rangle_w^1$ is linear in all variables.*

The first step to finding bounds (16) is to set $\mathbf{w} = (w)$, that is to take a single step, and find the weight function that makes the expectation $\langle q, f \rangle_w^1$ extremal.

Proposition 3 (Optimality principle). *Let $q, f: \mathcal{X} \rightarrow \mathbb{R}$ and w be some weight function. Set $h(x) = q(x)/W(x)$ for every $x \in \mathcal{X}$ and define*

$$\psi_{h,f}(x, y) = (h(x) - h(y))(f(y) - f(x)).$$

Then $\langle q, f \rangle_w^1 = \underline{\langle q, f \rangle}^1$ if

$$w(x, y) = \begin{cases} \underline{w}(x, y) & \text{if } \psi_{h,f}(x, y) > 0; \\ \overline{w}(x, y) & \text{if } \psi_{h,f}(x, y) \leq 0. \end{cases}$$

Proof. We prove the proposition by contradiction. Suppose that $\psi_{h,f}(x_0, y_0) > 0$ but $w(x_0, y_0) > \underline{w}(x_0, y_0)$. Let $0 < d \leq w(x_0, y_0) - \underline{w}(x_0, y_0)$ and set

$$w'(x, y) = \begin{cases} w(x, y) & \text{if } (x, y) \notin \{(x_0, y_0), (x_0, x_0), (y_0, y_0), (y_0, x_0)\}; \\ w(x, y) - d & \text{if } (x, y) \in \{(x_0, y_0), (y_0, x_0)\}; \\ w(x, y) + d & \text{if } (x, y) \in \{(x_0, x_0), (y_0, y_0)\}. \end{cases}$$

We have that

$$\begin{aligned} \langle q, f \rangle_{w'}^1 &= \sum_{x \in \mathcal{X}} q(x) \sum_{y \in \mathcal{X}} \frac{w'(x, y)}{W(x)} \\ &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{X}} w'(x, y) h(x) f(y) \\ &= \langle q, f \rangle_w^1 + d(h(x_0)f(x_0) + h(y_0)f(y_0) - h(x_0)f(y_0) - h(y_0)f(x_0)) \\ &= \langle q, f \rangle_w^1 - d\psi_{h,f}(x_0, y_0) \\ &< \langle q, f \rangle_w^1, \end{aligned}$$

which contradicts minimality of $\langle q, f \rangle_w^1$.

The case where $\psi_{h,f}(x_0, y_0) < 0$ is proved similarly. \square

Corollary 1. *Let $q, f \in \mathbb{R}^{\mathcal{X}}$ be arbitrary mappings. Then*

$$\min_{\mathbf{w} \in \mathcal{W}^n} \langle q, f \rangle_{\mathbf{w}}^n \quad (17)$$

is attained in a vector $\mathbf{w} = (w_1, \dots, w_n)$, where all w_i are extremal.

Proof. Let $\mathbf{w} = (w_1, \dots, w_n)$ be a vector minimizing (17). We will show that then there exists a vector \mathbf{w}' with extremal components such that $\langle q, f \rangle_{\mathbf{w}'}^n \geq \langle q, f \rangle_{\mathbf{w}}^n$.

Suppose that i th component w_i of \mathbf{w} is not extremal. Then set $\tilde{q} = qT_{(w_1, \dots, w_{i-1})}$ and $\tilde{f} = T_{(w_{i+1}, \dots, w_n)}$. It easily follows from Proposition 3 that the expression $\langle \tilde{q}, \tilde{f} \rangle_w^1$ is minimized by an extremal weight function w . Thus $\langle \tilde{q}, \tilde{f} \rangle_w^1 \geq \langle \tilde{q}, \tilde{f} \rangle_{w_i}^1 = \langle q, f \rangle_{\mathbf{w}}^n$. Thus replacing w_i with w in \mathbf{w} would still maximize (17). By repeating the same argument for all components with non-extremal weight functions, we confirm the corollary. \square

4.2 The structure of the set of extremal weights

The problem of calculating the extreme value and the optimal weights vector in the expression (17) is a high dimensional optimization problem. We need to find the optimal value of an nm dimensional real-valued vector where n stands for the number of time steps and m for the number of edges of the graph. Knowing that all feasible vectors of weights where the extremal values may be found is restricted to the extremal weight vectors brings the number of the values to be considered down to the set of all nm dimensional binary vectors. That gives 2^{mn} possibilities, which is in general still far more than a number tractable on any computer system. Moreover, it is not obvious how to endow this space with such a metric or norm structure that would help with the optimization. Therefore the aim of the rest of this paper is to give a heuristic approach to

finding reasonable approximations of the target extreme values and insights into the practical complexity of the problem, which is apparently too hard to be tackled exactly.

Yet there is an additional reduction of the complexity, which follows from the theoretical results in Section 4.1. Every component w_i of an extremal weight vector \mathbf{w} must be of the form w_ψ , where ψ is some function of the form $\psi_{h,f}$. In fact the sign of $\psi_{h,f}$ only matters. Clearly the sign of $\psi_{h,f}$ depends on the ordering of states induced by the maps h and f . In other words, every pair of orderings of states induces a feasible extremal weight function. That is, for every component we need to consider at most $(s!)^2$ extremal weight functions instead of 2^m of all possible extremal weight functions, where s denotes the number of states and m the number of edges in the graph. The number of feasible weights additionally decreases when taking into account the orderings of q and f , but still remains much too large to allow any kind of exhaustive examination. Therefore a reasonable approach is to start with some initial weight vector and try to improve it repeatedly until finding an optimal solution. Though, as we show in the continuation this process does not necessarily lead to a global optimum.

4.3 Finding local extrema

Corollary 1 gives a necessary condition for \mathbf{w} to give extremal value of $\langle q, f \rangle_{\mathbf{w}}^n$. However, there may be several weight vectors satisfying this condition, yielding different values. That is, we may have multiple local extrema due to non-convex nature of the problem. Let us illustrate this with an example.

Example 1. Let $\mathcal{X} = \{1, 2\}$,

$$\underline{w} = \begin{bmatrix} 0.1 & 0.2 \\ 0.2 & 0.1 \end{bmatrix}, \bar{w} = \begin{bmatrix} 0.8 & 0.9 \\ 0.9 & 0.8 \end{bmatrix},$$

$q = (1, 0)$, and $f = (0, 1)$. The marginal weight function is then $W = (1, 1)$. Our goal is to minimize $qT_{w_1}T_{w_2}f$, which in our case corresponds to the lower two step transition probability $\underline{P}(X_2 = 2|X_0 = 1)$.

Take

$$w = \begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}, w' = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix},$$

which are the only extremal weight functions. The transition operators corresponding to w and w' coincide with w and w' because of both marginal weights being equal to 1.

Now we have that $f_2 := T_w f = (0.9, 0.1)$ and $f'_2 := T_{w'} f = (0.2, 0.8)$. Hence

$$\text{sign}(\psi_{q,f_2}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad \text{sign}(\psi_{q,f'_2}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Hence the expression $qT_{w_1}f_2$ is minimized by taking $w_1 = w$ and similarly the expression $qT_{w'_1}f'_2$ is minimized by taking $w'_1 = w'$.

To make things clearer we will parametrize all two-step weight vectors. Let

$$w_1(\alpha) = \alpha w + (1 - \alpha)w' \quad \text{and} \quad w_2(\beta) = \beta w + (1 - \beta)w'.$$

Now we can express explicitly

$$F(\alpha, \beta) = qT_{w_1(\alpha)}T_{w_2(\beta)}f = 0.32 + 0.42\alpha + 0.42\beta - 0.98\alpha\beta,$$

whose local extrema in $[0, 1]^2$ are $F(0, 0) = 0.32$ and $F(1, 1) = 0.18$ which are local minima while in the local maxima in $(0, 1)$ and $(1, 0)$ the same value 0.74 is attained. The only stationary point in the interior of $[0, 1]^2$ is a saddle point and thus not a local extreme.

Having characterized local extrema, we can now provide a simple Algorithm 1 that finds local extrema from some starting weight vector \mathbf{w} . We have

Algorithm 1 FindLocalMinimum

```

1: function LOCALMINIMUM( $q, initialWeightVector, f$ )
2:    $weightVector \leftarrow initialWeightVector$ 
3:   repeat
4:      $k \leftarrow setSplitPoint$  ▷ set the division point
5:      $wl \leftarrow weightVector(1 : k - 1)$  ▷ left part of the weight vector
6:      $wr \leftarrow weightVector(k + 1 : n)$  ▷ right part of the weight vector
7:      $ql \leftarrow qT_{wl}$ 
8:      $fr \leftarrow T_{wr}f$ 
9:      $wnew \leftarrow w_{\psi_{ql, fr}}$  ▷ set the weight function  $w$  that minimizes  $qlT_wfr$ 
10:     $weightVector \leftarrow (wl, wnew, wr)$ 
11:  until there are no more division points where  $qlT_{w_k}fr$  is not minimal
12: end function

```

left the way how the split point is selected open on purpose in Algorithm 1, line 4, because there are many ways how we can proceed with this. Thus, we can for instance start from left to right and repeat the process until we find no more locally non-optimal weight functions. Or we might go the other way around. The order does affect the results. That is, not only that the number of iterations needed may be different, but also the resulting locally optimal weights vector \mathbf{w} may be different.

4.4 Testing the local algorithm

It takes just a slight modification of Algorithm 1 to find local maxima instead of minima. To understand its efficiency for finding global extrema, we implemented some numerical testing with random interval weights in graphs of various sizes to answer the following relevant questions:

1. How many unique local extrema does typical problem of the form (16) have?
2. How does the order of split points affect the resulting local extreme?
3. Can the value of $\langle q, f \rangle_{w_{initial}}^n$ in any way predict the value in the resulting local extreme?
4. Does the value in a local extreme affect the likelihood of the algorithm resulting in that extreme?

		time steps		
		2	4	6
vertices	4	1.9	13.4	80.6
	6	3.2	46.8	251.2
	8	5.3	100.3	411.4

Table 1: Average number of local extrema.

We have tested the algorithm on graphs with 4, 6 and 8 vertices. Lower weights $\underline{w}(x, y)$ were randomly generated using exponential distribution with expected value 0.8 and the upper bounds were generated as $\overline{w}(x, y) = \underline{w}(x, y)X$ where X were random numbers distributed exponentially with the expected value 1. Entries of both q and f were generated as random numbers distributed exponentially with expected value 1.5. Each graph contained about 1/4 of pairs of vertices that were not connected.

For each set of parameters, i.e. $\underline{w}, \overline{w}, q$ and f , a sample of 1500 extremal weight vectors was generated using random permutations (see the last part of Section 4.2) and then calculated the corresponding local extreme using Algorithm 1. We tested for random walks of lengths 2, 4 and 6 respectively. For each size of graphs and random walk length a sample of 200 sets of parameters was generated.

Typical distribution of the number of (discovered) extreme points is best modelled with exponential distribution with the means that depend on the number of vertices and time steps. The parameters are listed in Table 1. Further we tested the influence of the order in which non-optimal weight functions are selected and optimized in Algorithm 1, line 4. We tested left-to-right and right-to-left order. It turns out that most of the time the resulting local extrema do not coincide. However, the comparison of overall frequencies of the obtained local extrema do not show any systematical differences in their distributions. In Figure 1 frequencies of local minima and maxima respectively are depicted for the left-to-right and right-to-left orders. One can observe similar distributions for both orders.

One might expect that the starting weight function can give some information about the local extreme it leads to. It could be for instance that low value in the starting point predicts lower value in the resulting local minimum, which could be used to do preliminary selection of the starting points. However, empirical results show no significant correlation between the two values. A graph showing dependency between the initial and optimized value is shown in Figure 2. It clearly shows that the initial value does not reveal much information about the corresponding local maximum. Although the efficiency of the algorithm can be clearly observed by comparing the initial and optimized values. Experimental results neither show any particular general pattern that would suggest that the value of a local extreme would impact the probability to be attained by the local optimization algorithm.

4.5 From local to global extrema

So far the best way to find global extrema for random walks in weighted graphs is to take some sample of random initial weight vectors, do the local optimiza-

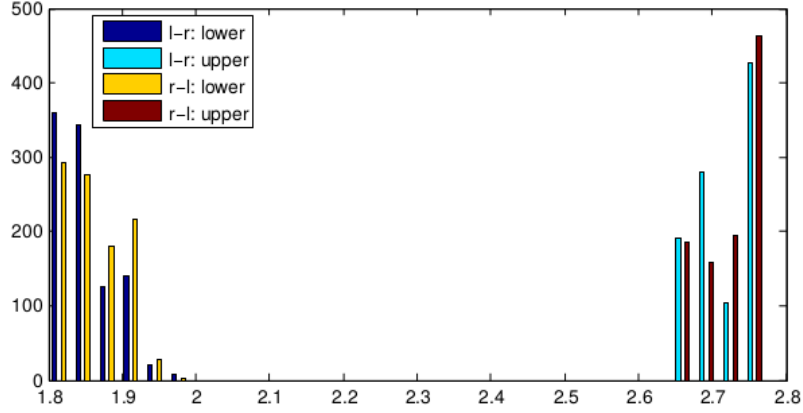


Figure 1: Frequencies of left-to-right and right-to-left order in Algorithm 1 for lower and upper bounds.

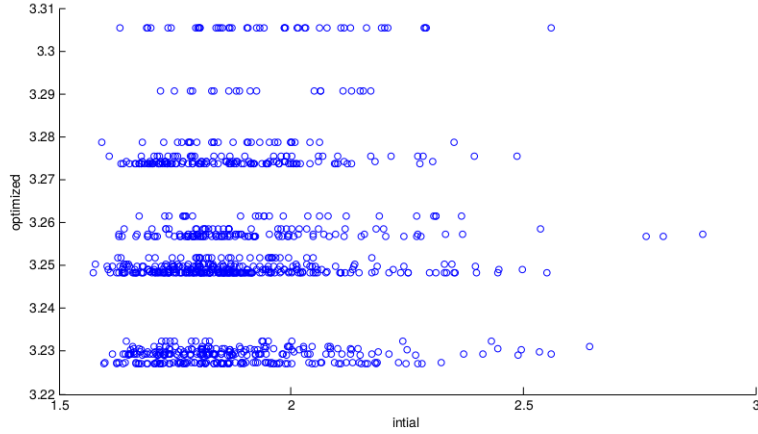


Figure 2: Comparison of the initial value of the maximizing function and the value after applying the local optimization algorithm.

tion, and hope that one of the so obtained local extrema is a global extrema. This method, however, does not contain any decisive criterion whether the best obtained solution is globally best solution. It is therefore not clear how big sample one has to take in order to get an estimate reasonably close to the true optimal solution with reasonable certainty. In most cases not too large samples are needed for this. We have generated a sample of 300 parameter sets for interval weighted graphs with 8 vertices and the same number of time steps. All q and f were non-negative to ensure non-negativity of the resulting optima and therefore comparison of relative deviations. Both interval weights and widths were generated as exponentially distributed random values. For each parameter set a sample of 2500 randomly generated initial points was generated and then optimized to get the corresponding local extrema. We have calculated the values $\langle g, f \rangle_{\mathbf{w}}^n$ for the initial extremal vector of weights \mathbf{w} and for the optimized weight vector \mathbf{w}_{opt} . In Figure 3 the average relative deviation, in percentage of the

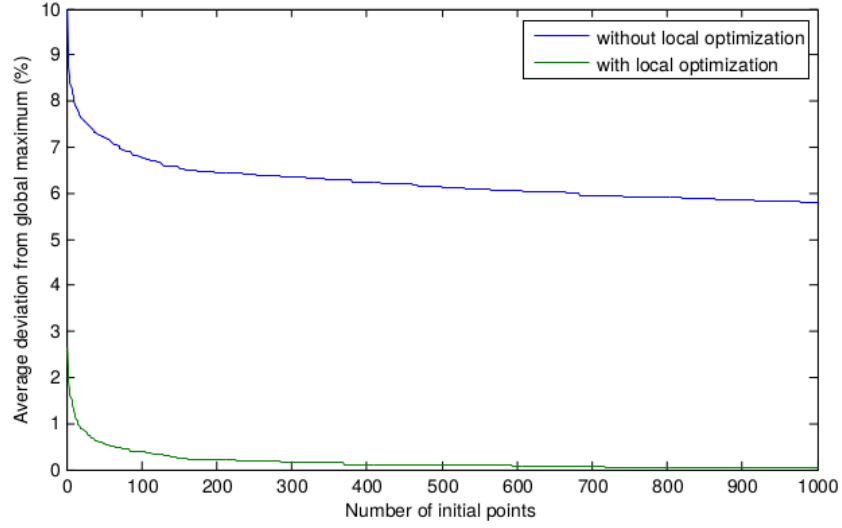


Figure 3: Average relative deviations from optimal solution for given sample of initial points with and without local optimization.

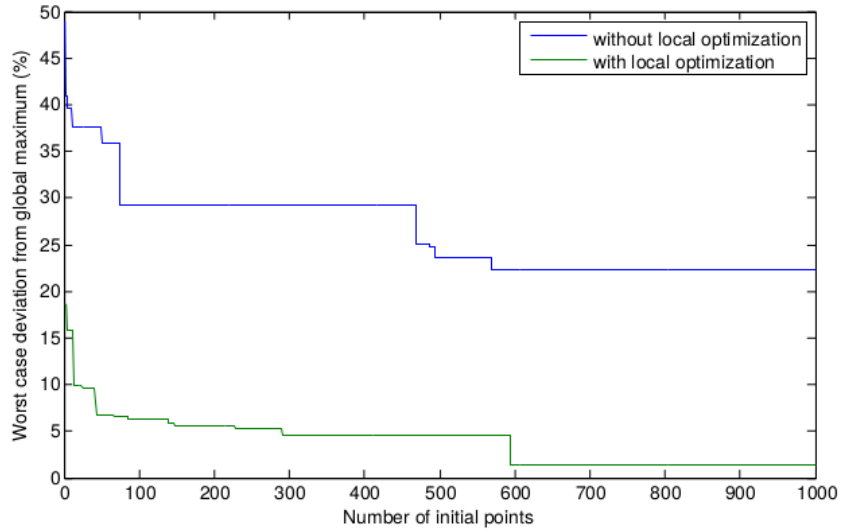


Figure 4: Maximal relative deviations from optimal solution for given sample of initial points with and without local optimization.

best solution, from the most optimal value found is graphed for initial and optimized vector weights depending on the size of the sample. In Figure 4 the worst case (maximal) relative deviation is graphed depending on the sample size. It can be clearly observed that locally optimized solutions by far outperform best solutions that might be found by randomly generated extremal weights only. Secondly, we can also observe that even if it cannot be guaranteed that best so-

lution has been found among locally optimized solutions, the deviations become reasonable at not too big sample sizes.

5 Conclusion and further work

As far as we are aware this paper is a first attempt to model random walks on weighted graphs with interval weights, and also reversible imprecise Markov chains. We have addressed the most basic question about calculating transition probabilities for multiple time steps. This is a non-convex problem whose computational complexity grows exponentially with the number of time steps. Thanks to the local optimization algorithm we can find reasonable approximations of global optima with a tractable amount of computation.

Our approach only works in the case where marginals are known precisely. It is therefore natural to try to extend calculations without the assumption of fixed sum of weights and even to allow more general sets of weight functions besides those which are described in terms of intervals. While both possibilities would be plausible, computational complexity may be an obstacle to prevent their efficient analysis.

Regarding long term distribution our model is a simple one with known limit distribution. Although there are still several relevant questions to be addressed, such as computing mixing times, cover and hitting times.

References

- [1] Abe, Yoshihiro. 2014. Cover times for sequences of reversible Markov chains on random graphs. *Kyoto J. Math.*, **54**(3), 555–576.
- [2] Aissi, Hassene, Bazgan, Cristina, & Vanderpooten, Daniel. 2009. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European journal of operational research*, **197**(2), 427–438.
- [3] Aldous, David, & Fill, James Allen. 2002. *Reversible Markov Chains and Random Walks on Graphs*. Unfinished monograph, recompiled 2014, available at <http://www.stat.berkeley.edu/~aldous/RWG/book.html>.
- [4] Aldous, David J. 1989. Lower bounds for covering times for reversible Markov chains and random walks on graphs. *Journal of Theoretical Probability*, **2**(1), 91–100.
- [5] Augustin, Thomas, Coolen, Frank PA, de Cooman, Gert, & Troffaes, Matthias CM. 2014. *Introduction to imprecise probabilities*. John Wiley & Sons.
- [6] Averbakh, Igor, & Lebedev, Vasilij. 2004. Interval data minmax regret network optimization problems. *Discrete Applied Mathematics*, **138**(3), 289–301.
- [7] Backstrom, Lars, & Leskovec, Jure. 2011. Supervised random walks: predicting and recommending links in social networks. *Pages 635–644 of: Proceedings of the fourth ACM international conference on Web search and data mining*. ACM.

- [8] Coppersmith, Don, Doyle, Peter, Raghavan, Prabhakar, & Snir, Marc. 1993. Random walks on weighted graphs and applications to on-line algorithms. *Journal of the ACM (JACM)*, **40**(3), 421–453.
- [9] de Cooman, Gert, Hermans, Filip, & Quaeghebeur, Erik. 2009. Imprecise Markov chains and their limit behavior. *Probability in the Engineering and Informational Sciences*, **23**(4), 597–635.
- [10] Feige, Uriel. 1995a. A tight lower bound on the cover time for random walks on graphs. *Random Structures and Algorithms*, **6**(4), 433–438.
- [11] Feige, Uriel. 1995b. A tight upper bound on the cover time for random walks on graphs. *Random Structures and Algorithms*, **6**(1), 51–54.
- [12] Fouss, Francois, Pirotte, Alain, Renders, Jean-Michel, & Saerens, Marco. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *Knowledge and data engineering, IEEE transactions on*, **19**(3), 355–369.
- [13] Göbel, F, & Jagers, AA. 1974. Random walks on graphs. *Stochastic processes and their applications*, **2**(4), 311–336.
- [14] Green, Peter J. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**(4), 711–732.
- [15] Hartfiel, Darald J. 1998. *Markov Set-Chains*. Berlin: Springer-Verlag.
- [16] Hastings, W Keith. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**(1), 97–109.
- [17] Jerrum, Mark, & Sinclair, Alistair. 1996. The Markov chain Monte Carlo method: an approach to approximate counting and integration. *Approximation algorithms for NP-hard problems*, 482–520.
- [18] Li, Rong-Hua, Yu, Jeffrey Xu, & Liu, Jianquan. 2011. Link prediction: the power of maximal entropy random walk. *Pages 1147–1156 of: Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM.
- [19] Lin, Yuan, & Zhang, Zhongzhi. 2014. Mean first-passage time for maximal-entropy random walks in complex networks. *Scientific reports*, **4**.
- [20] Liu, Hongyu, & Milios, Evangelos. 2012. Probabilistic models for focused web crawling. *Computational Intelligence*, **28**(3), 289–328.
- [21] Lovász, László. 1993. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty*, **2**(1), 1–46.
- [22] Miranda, Enrique. 2008. A survey of the theory of coherent lower previsions. *International Journal of Approximate Reasoning*, **48**(2), 628 – 658. In Memory of Philippe Smets (1938-2005).

- [23] Pons, Pascal, & Latapy, Matthieu. 2005. Computing Communities in Large Networks Using Random Walks. *Pages 284–293 of: Yolum, pInar, Gngr, Tunga, Grgen, Fikret, & zturan, Can (eds), Computer and Information Sciences - ISCIS 2005*. Lecture Notes in Computer Science, vol. 3733. Springer Berlin Heidelberg.
- [24] Puterman, Martin L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
- [25] Škulj, Damjan. 2009. Discrete time Markov chains with interval probabilities. *International Journal of Approximate Reasoning*, **50**(8), 1314–1329.
- [26] Škulj, Damjan. 2015. Efficient computation of the bounds of continuous time imprecise Markov chains. *Applied Mathematics and Computation*, **250**(0), 165 – 180.
- [27] Škulj, Damjan, & Hable, Robert. 2013. Coefficients of ergodicity for Markov chains with uncertain parameters. *Metrika*, **76**(1), 107–133.
- [28] Yin, Zhijun, Gupta, Manish, Weninger, Tim, & Han, Jiawei. 2010. A unified framework for link recommendation using random walks. *Pages 152–159 of: Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*. IEEE.
- [29] Zhang, Zhongzhi, Li, Huan, & Sheng, Yibin. 2014. Effects of reciprocity on random walks in weighted networks. *Scientific reports*, **4**.